



# Availability Analysis and Stationary Regime Detection of Markov Processes

Bruno Sericola

## ► To cite this version:

Bruno Sericola. Availability Analysis and Stationary Regime Detection of Markov Processes. [Research Report] RR-2886, INRIA. 1996. inria-00073804

**HAL Id: inria-00073804**

**<https://inria.hal.science/inria-00073804>**

Submitted on 24 May 2006

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

***Availability analysis and stationary regime  
detection of Markov processes***

Bruno Sericola

**N° 2886**

Mai 1996

\_\_\_\_\_ THÈME 1 \_\_\_\_\_

 ***apport  
de recherche***



# Availability analysis and stationary regime detection of Markov processes

Bruno Sericola

Thème 1 — Réseaux et systèmes  
Projet Model

Rapport de recherche n° 2886 — Mai 1996 — 17 pages

**Abstract:** Point availability and expected interval availability are dependability measures respectively defined by the probability that a system is in operation at a given instant and by the mean percentage of time during which a system is in operation over a finite observation period. We consider a repairable computer system and we assume as usual that the system is modeled by a finite Markov process. We propose in this paper a new algorithm to compute these two availability measures. This algorithm is based on the classical uniformization technique in which a test to detect the stationary behavior of the system is used to stop the computation if the stationarity is reached. In that case, the algorithm gives not only the transient availability measures but also the steady state availability, with significant computational savings especially when the time at which measures are needed is large. In the case where the stationarity is not reached, the algorithm provides the transient availability measures and bounds for the steady state availability. It is also shown how the new algorithm can be extended to the computation of performability measures.

**Key-words:** Repairable computer systems, dependability, availability, performability, Markov processes, stationarity detection.

*(Résumé : tsvp)*

# Analyse de disponibilité et détection du régime stationnaire des processus markoviens

**Résumé :** La disponibilité ponctuelle et la disponibilité moyenne sur un intervalle sont des mesures de sûreté de fonctionnement respectivement définies par la probabilité qu'un système soit opérationnel à un instant donné et par le pourcentage de temps moyen durant lequel le système est opérationnel sur une période d'observation finie. On considère un système informatique réparable et l'on suppose, comme souvent, que le système est modélisé par un processus markovien fini. On propose dans cet article un nouvel algorithme pour calculer ces deux mesures de disponibilité. Cet algorithme est basé sur la technique classique de l'uniformisation dans laquelle un test détectant le comportement stationnaire du système est utilisé pour arrêter le calcul si le régime stationnaire est atteint. Dans ce cas, l'algorithme fournit non seulement les mesures de disponibilité transitoires mais aussi la disponibilité stationnaire, avec une économie de calculs significative, en particulier lorsque l'instant auquel les mesures sont demandées est important. Dans le cas où le régime stationnaire n'est pas détecté, l'algorithme fournit les mesures transitoires et des bornes pour la disponibilité stationnaire. On montre aussi comment ce nouvel algorithme peut être étendu au calcul de mesures de performabilité.

**Mots-clé :** Système informatique réparable, sûreté de fonctionnement, disponibilité, performabilité, processus markovien, détection du régime stationnaire.

# 1 Introduction

In the dependability analysis of repairable computing systems, there is an increasing interest in evaluating transient measures, in particular, the point availability and the availability over a given period. This paper deals with the computation of the point availability and of the expected interval availability respectively defined by the probability that the system is in operation at a given instant and by the mean percentage of time during which the system is in operation over a finite observation period. Formally, the system is modeled by a Markov process. Its state space is divided into two disjoint sets which represent the *up* states in which the system delivers the specified service and the *down* states in which there is no more service delivered. Transitions from the *up* (resp. *down*) states to the *down* (resp. *up*) states are called *failures* (resp. *repairs*). The interval availability over  $(0, t)$  is then the fraction of the interval  $(0, t)$  during which the process is in the *up* states. This random variable has been studied in previous papers as for instance in [1], [2] and [3] where its distribution is evaluated using the uniformization technique. This approach is interesting because it has good numerical properties and it allows the user to perform the computation with an error as small as desired.

An approach to detect the stationarity of Markov processes has been proposed in [4], [5]. This approach is based on the uniformization method. The state probability vectors of the uniformized Markov chain are successively computed and the iterates that are spaced  $m$  iterations apart are compared. When the difference between two such iterates is small enough, the computation is stopped. The main problem with this method is that, unlike the standard uniformization, there is no ability to specify error bounds easily computable.

In this paper, we develop a new method to compute the point availability and the expected interval availability which is also based on the uniformization technique and on the stationary regime detection. In practice one usually does not know whether the time horizon he/she is considering is large enough for a steady state analysis. The main advantage of our algorithm is that the computation is stopped when the steady state availability of the system is reached giving both transient and steady state measures with an error tolerance specified in advance. When the stationarity is not reached, the algorithm gives the transient measures and bounds for the steady state availability.

The remainder of the paper is organized as follows. In the following section, we recall the classical way to compute the point availability and we derive new results to stop the computation when the stationary regime is reached. We also give in this section the pseudo code of both algorithms. In Section 3, we consider the expected interval availability and we show how it can be computed using the stationarity detection. In Section 4, we show by means of a numerical example that our new algorithm can considerably reduce the computation time of the availability measures considered here, when the time at which measures are needed is sufficiently large. It is also shown that computational savings

can be obtained even when the time horizon is small. In Section 5, we show how the results obtained for the availability measures can be easily extended to the corresponding performability measures. The last section is devoted to some conclusions.

## 2 Point Availability Analysis

Consider an irreducible continuous-time homogeneous Markov process  $X = \{X_t, t \geq 0\}$ , over a finite state space denoted by  $S$ . The states of  $S$  are divided into two disjoint subsets:  $U$ , the set of the operational states (or the up states) and  $D$ , the set of the unoperational states (or the down states). For a system, modeled by such a process, the point availability at time  $t$  is denoted by  $PAV(t)$  and defined by

$$PAV(t) = \Pr\{X_t \in U\}.$$

The process  $X$  is, as usual, given by its infinitesimal generator, denoted by  $A$ , in which the  $i$ th diagonal entry  $A(i, i)$  verifies  $A(i, i) = -\sum_{j \neq i} A(i, j)$ . Its initial probability distribution is denoted by the row vector  $\alpha$ .

The uniformized Markov chain associated to the process  $X$  is characterized by its uniformization rate  $\nu$  and by its transition probability matrix  $P$  [6]. The uniformization rate  $\nu$  verifies  $\nu \geq \max(-A(i, i); i \in S)$  and  $P$  is related to  $A$  by  $P = I + A/\nu$ , where  $I$  denotes the identity matrix. Using this notation, we get

$$PAV(t) = \alpha e^{At} 1_U = \sum_{n=0}^{+\infty} e^{-\nu t} \frac{(\nu t)^n}{n!} \alpha P^n 1_U, \quad (1)$$

where  $1_U$  is a column vector whose  $i$ th entry is 1 if  $i \in U$ , and 0 if  $i \in D$ . We denote by  $V_n$  the column vector defined by  $V_n = P^n 1_U$ . It follows that for every  $n \geq 0$ , we have  $V_n = P V_{n-1}$  and  $V_0 = 1_U$ . In the following, we define for every  $n \geq 0$ ,  $v_n = \alpha P^n 1_U = \alpha V_n$ .

### 2.1 The classical uniformization method

The classical way to compute the point availability at time  $t$  is based on Relation (1). Let  $\varepsilon$  be a given specified error tolerance and  $N$  be defined as

$$N = \min \left\{ n \in \mathbb{N} \mid \sum_{j=0}^n e^{-\nu t} \frac{(\nu t)^j}{j!} \geq 1 - \varepsilon \right\}. \quad (2)$$

Then we obtain

$$PAV(t) = \sum_{n=0}^N e^{-\nu t} \frac{(\nu t)^n}{n!} v_n + e(N),$$

where the rest of the series  $e(N)$  verifies

$$e(N) = \sum_{n=N+1}^{\infty} e^{-\nu t} \frac{(\nu t)^n}{n!} v_n \leq \sum_{n=N+1}^{\infty} e^{-\nu t} \frac{(\nu t)^n}{n!} = 1 - \sum_{n=0}^N e^{-\nu t} \frac{(\nu t)^n}{n!} \leq \varepsilon.$$

The computation of integer  $N$  can be made without any numerical problems even for large values of  $\nu t$  by using the method described in [7].

The truncation level  $N$  is in fact a function of  $t$ , say  $N_t$ . For a fixed value of  $\varepsilon$ ,  $N_t$  is an increasing function of  $t$ . It follows that if we want to compute  $PAV(t)$  for  $J$  distinct values of  $t$ , denoted by  $t_1 < \dots < t_J$ , we only need to compute  $v_n$  for  $n = 1, \dots, N_{t_J}$  since the values of  $v_n$  are independent of the parameter  $t$ .

The pseudo code of the classical uniformization method can then be written as follows.

```

input :  $\varepsilon, t_1 < \dots < t_J$ 
output :  $PAV(t_1), \dots, PAV(t_J)$ 
Compute  $N$  from Relation (2) with  $t = t_J$ 
 $V_0 = 1_U; v_0 = \alpha V_0$ 
for  $n = 1$  to  $N$  do
     $V_n = P V_{n-1}; v_n = \alpha V_n$ 
endfor
for  $j = 1$  to  $J$  do
     $PAV(t_j) = \sum_{n=0}^N e^{-\nu t_j} \frac{(\nu t_j)^n}{n!} v_n$ 
endfor

```

Table 1: Classical algorithm for the computation of  $PAV(t)$ .

## 2.2 Stationarity detection

The stationarity detection that we consider is based on the control of the sequence of vectors  $V_n = P^n 1_U$ . Let the row vector  $\pi$  denote the stationary probability distribution of the Markov process  $X$ . This vector verifies  $\pi A = 0$  and  $\pi P = \pi$ . The steady state availability is given by  $PAV(\infty) = \pi 1_U$ . To ensure the convergence of the sequence of vectors  $V_n$ , we require that the uniformization rate  $\nu$  verifies  $\nu > \max(-A(i, i); i \in S)$  since this guarantees that the transition probability matrix  $P$  is aperiodic. We then have, for every  $i \in S$ ,

$$\lim_{n \rightarrow \infty} V_n(i) = \pi 1_U.$$

We describe now the test used to detect that, for a given value of  $n$ , the entries of vector  $V_n$  are close to  $\pi 1_U$ . For every  $n \geq 0$ , we define

$$m_n = \min_{i \in S} V_n(i) \quad \text{and} \quad M_n = \max_{i \in S} V_n(i).$$



Note that, since  $V_0 = 1_U$ , we have  $M_0 = 1$  and  $m_0 = 0$ . The following result gives bounds of the steady state availability  $PAV(\infty) = \pi 1_U$ .

**Lemma 2.1** *The sequences  $m_n$  and  $M_n$  are respectively non decreasing and non increasing and, for every  $n \geq 0$ , we have*

$$\left| v_n - \frac{M_n + m_n}{2} \right| \leq \frac{M_n - m_n}{2} \text{ and } \left| \pi 1_U - \frac{M_n + m_n}{2} \right| \leq \frac{M_n - m_n}{2}.$$

Moreover, both sequences  $m_n$  and  $M_n$  converge to  $\pi 1_U$ .

**Proof.** For every  $i \in S$ , we have  $V_{n+1}(i) = \sum_{j \in S} P(i, j) V_n(j)$ . It follows that  $m_n \leq V_{n+1}(i) \leq M_n$  and so we get  $m_n \leq m_{n+1}$  and  $M_{n+1} \leq M_n$ , which shows that the sequences  $m_n$  and  $M_n$  are respectively non decreasing and non increasing.

Since  $v_n = \sum_{j \in S} \alpha(j) V_n(j)$ , we get  $m_n \leq v_n \leq M_n$ , which is equivalent to

$$\left| v_n - \frac{M_n + m_n}{2} \right| \leq \frac{M_n - m_n}{2}.$$

Writing now  $\pi 1_U = \pi P^n 1_U = \pi V_n = \sum_{j \in S} \pi(j) V_n(j)$ , we get in the same way  $m_n \leq \pi 1_U \leq M_n$ , which is equivalent to

$$\left| \pi 1_U - \frac{M_n + m_n}{2} \right| \leq \frac{M_n - m_n}{2}.$$

It is shown in [8], page 270, that

$$\lim_{n \rightarrow \infty} \min_{i \in S} P^n(i, j) = \lim_{n \rightarrow \infty} \max_{i \in S} P^n(i, j) = \pi(j).$$

The proof of this result can be easily adapted to our problem, to show that both sequences  $m_n$  and  $M_n$  converge to  $\pi 1_U$ . ■

This lemma shows that the difference  $M_n - m_n$  converges to 0, that is, for a fixed error tolerance  $\varepsilon > 0$  there exists an integer  $k$  such that for  $n \geq k$  we have  $M_n - m_n \leq \varepsilon$ . Since  $m_n \leq M_n$ , we have  $m_n \leq m_{n+1} \leq M_{n+1} \leq M_n$ , and so the sequence  $(M_n - m_n)$  is non increasing. We can then define the following integer

$$K = \inf\{n \geq 0 | M_n - m_n \leq \varepsilon/2\}.$$

Using the integer  $K$ , Relation (1) can be written as

$$PAV(t) = \sum_{n=0}^K e^{-\nu t} \frac{(\nu t)^n}{n!} v_n + \frac{M_K + m_K}{2} \left( 1 - \sum_{n=0}^K e^{-\nu t} \frac{(\nu t)^n}{n!} \right) + e_1(K), \quad (3)$$

where

$$e_1(K) = \sum_{n=K+1}^{\infty} e^{-\nu t} \frac{(\nu t)^n}{n!} v_n - \frac{M_K + m_K}{2} \sum_{n=K+1}^{\infty} e^{-\nu t} \frac{(\nu t)^n}{n!}.$$

Using Lemma 2.1, the rest  $e_1(K)$  verifies

$$|e_1(K)| \leq \sum_{n=K+1}^{\infty} e^{-\nu t} \frac{(\nu t)^n}{n!} \left| v_n - \frac{M_K + m_K}{2} \right| \leq \varepsilon/4. \quad (4)$$

This last inequality follows from the fact that, for  $n \geq K$ , we have from Lemma 2.1  $m_K \leq m_n \leq v_n \leq M_n \leq M_K$  and so  $\left| v_n - \frac{M_K + m_K}{2} \right| \leq \frac{M_K - m_K}{2} \leq \varepsilon/4$ .

The time  $K$  can be interpreted as the discrete time to stationarity with respect to the subset  $U$ .

For every  $t \geq 0$  and for every integer  $l \geq 0$ , we denote by  $F_l(t)$  the function defined by

$$F_l(t) = \sum_{n=0}^l e^{-\nu t} \frac{(\nu t)^n}{n!} (M_n - m_n).$$

It is easy to check that for a fixed value of  $l$ , the function  $F_l(t)$  decreases, from 1 to 0 over the interval  $[0, \infty[$ . We can then define for every integer  $l \geq 0$  and for every  $\varepsilon > 0$ , the time  $T_l$  as

$$T_l = \inf\{t \geq 0; F_l(t) \leq \varepsilon/4\}.$$

We then have the following theorem :

**Theorem 2.2** *For every  $\varepsilon > 0$ , for every  $t \geq T_K$  we have*

$$|PAV(t) - \pi 1_U| \leq 3\varepsilon/4 \quad (5)$$

$$\left| \pi 1_U - \frac{M_K + m_K}{2} \right| \leq \varepsilon/4 \quad (6)$$

$$\left| PAV(t) - \frac{M_K + m_K}{2} \right| \leq \varepsilon \quad (7)$$

**Proof.** First note that, from Lemma 2.1, we have  $m_n \leq v_n \leq M_n$  and  $m_n \leq \pi 1_U \leq M_n$ , for every  $n \geq 0$ . It follows that  $|v_n - \pi 1_U| \leq M_n - m_n$  for every  $n \geq 0$ . We then have

$$\begin{aligned} |PAV(t) - \pi 1_U| &= \left| \sum_{n=0}^{\infty} e^{-\nu t} \frac{(\nu t)^n}{n!} v_n - \pi 1_U \right| \\ &\leq \sum_{n=0}^{\infty} e^{-\nu t} \frac{(\nu t)^n}{n!} |v_n - \pi 1_U| \\ &\leq \sum_{n=0}^{\infty} e^{-\nu t} \frac{(\nu t)^n}{n!} (M_n - m_n) \\ &= F_K(t) + \sum_{n=K+1}^{\infty} e^{-\nu t} \frac{(\nu t)^n}{n!} (M_n - m_n) \end{aligned}$$

Since  $t \geq T_K$ , we have  $F_K(t) \leq \varepsilon/4$ . In the second term, since  $n \geq K$ , we have  $M_n - m_n \leq M_K - m_K \leq \varepsilon/2$  and so we get Relation (5).

Relation (6) is immediate from Lemma 2.1. Finally combining Relation (5) and Relation (6), we get Relation (7). ■

The time  $T_K$  can be interpreted as the continuous time to stationarity with respect to the subset  $U$ .

## 2.3 The new algorithm

Using these results, we obtain the following new algorithm. To simplify the writing of this algorithm, we define

$$G_l(t) = \sum_{n=0}^l e^{-\nu t} \frac{(\nu t)^n}{n!} v_n, \quad H_l(t) = 1 - \sum_{n=0}^l e^{-\nu t} \frac{(\nu t)^n}{n!}, \quad S_l = \frac{M_l + m_l}{2}.$$

```

input :  $\varepsilon, t_1 < \dots < t_J$ 
output :  $PAV(t_1), \dots, PAV(t_J)$ 
Compute  $N$  from Relation (2) with  $t = t_J$ 
 $V_0 = 1_U; v_0 = \alpha V_0$ 
 $M_0 = 1; m_0 = 0; K = N + 1$ 
for  $n = 1$  to  $N$  do
     $V_n = P V_{n-1}; v_n = \alpha V_n$ 
    Compute  $M_n, m_n$  and  $S_n$ 
    if  $(M_n - m_n \leq \varepsilon/2)$ 
         $K = n; \text{break}$ 
    endif
endfor
if  $(K = N + 1)$ 
    for  $j = 1$  to  $J$  do  $PAV(t_j) = G_N(t_j)$  endfor
     $m_N \leq PAV(\infty) \leq M_N$ 
endif
if  $(K \leq N)$ 
    Compute  $T_K = \inf\{t \geq 0; F_K(t) \leq \varepsilon/4\}$ 
    for  $j = 1$  to  $J$  do
        if  $(t_j \leq T_K)$  then  $PAV(t_j) = G_K(t_j) + S_K H_K(t_j)$ 
        if  $(t_j > T_K)$  then  $PAV(t_j) = PAV(\infty) = S_K$ 
        endif
    endfor
endif

```

Table 2: Algorithm for the computation of  $PAV(t)$  using stationarity detection.

Note that it is not necessary to compute the continuous time to stationarity  $T_K$  with a high precision. It is sufficient to obtain an upper bound of  $T_K$  such as for instance  $\lceil T_K \rceil$  which is the smallest integer greater or equal to  $T_K$ .

It must be also noted that, in this algorithm, the truncation step  $N$  is a function of the time  $t_J$  as in the classical uniformization algorithm but the times to stationarity  $K$  and  $T_K$  are independent of the time parameter, when the discrete time  $K$  is reached.

The computational time complexity of both algorithm is essentially due to the computation of the vectors  $V_n$ . To compute this vectors, the classical algorithm requires  $N$  matrix-vector products and our new algorithm requires only  $\min(K, N)$  matrix-vector products.

### 3 Expected Interval Availability Analysis

We show in this section how the new algorithm proposed above for the point availability computation can be adapted to compute the expected interval availability taking account of the stationarity detection.

The expected interval availability represents the mean percentage of time during which the system is in operation over a finite observation period  $(0, t)$ . The interval availability over  $(0, t)$  is denoted by  $IAV(t)$  and its expectation is given by

$$EIAV(t) = \frac{1}{t} \int_0^t PAV(s) ds.$$

Using Relation (1) and by integration over  $(0, t)$ , we obtain

$$EIAV(t) = \sum_{n=0}^{+\infty} e^{-\nu t} \frac{(\nu t)^n}{n!} \frac{1}{n+1} \sum_{k=0}^n \alpha P^k 1_U.$$

We denote by  $V'_n$  the column vector defined by

$$V'_n = \frac{1}{n+1} \sum_{k=0}^n P^k 1_U,$$

and we define  $v'_n = \alpha V'_n$ . By definition of  $V_n$  and  $v_n$  in the previous section, we get, for every  $n \geq 0$ ,

$$V'_n = \frac{1}{n+1} \sum_{k=0}^n V_k \text{ and } v'_n = \frac{1}{n+1} \sum_{k=0}^n v_k.$$

It follows that  $V'_n$  and  $v'_n$  are recursively given, for  $n \geq 1$ , by

$$V'_n = \frac{n}{n+1} V'_{n-1} + \frac{1}{n+1} V_n,$$

and

$$v'_n = \frac{n}{n+1}v'_{n-1} + \frac{1}{n+1}v_n, \quad (8)$$

with  $V'_0 = V_0 = 1_U$  and thus  $v'_0 = v_0$ . For every  $n \geq 0$ , we have  $0 \leq v'_n \leq 1$ . It follows that, using the truncation step  $N$  defined in Relation (2), we get the classical algorithm to compute the expected interval availability, by writing

$$EIAV(t) = \sum_{n=0}^N e^{-\nu t} \frac{(\nu t)^n}{n!} v'_n + e'(N),$$

where

$$e'(N) = \sum_{n=N+1}^{\infty} e^{-\nu t} \frac{(\nu t)^n}{n!} v'_n \leq \sum_{n=N+1}^{\infty} e^{-\nu t} \frac{(\nu t)^n}{n!} = 1 - \sum_{n=0}^N e^{-\nu t} \frac{(\nu t)^n}{n!} \leq \varepsilon.$$

This algorithm is basically as the one depicted in Table 1. More precisely the computation of  $v_n$  in Table 1 must be followed by the recursion (8), with  $v'_0 = v_0$ , and in the last loop over  $j$ ,  $v_n$  must be replaced by  $v'_n$  in order to get  $EIAV(t_j)$  instead of  $PAV(t_j)$ .

### 3.1 Stationarity detection for the expected interval availability

Using the results obtained for the point availability, we can derive a new method to obtain the expected interval availability using the stationarity detection. This method is based on the two following theorems. Both theorems will be used in the case where the discrete time to stationarity  $K$  is such that  $K \leq N$ . The first theorem states that in order to compute the expected interval availability,  $EIAV(t)$ , we only need the values of  $v'_n$  for  $n \leq K$ . The second theorem states that in order to compute the expected interval availability,  $EIAV(t)$ , for  $t \geq T_K$ , we only need the value  $EIAV(t')$  at a time  $t'$  such that  $t \geq t' \geq T_K$ .

We denote by  $G'_K(t)$  the function

$$G'_K(t) = \sum_{n=0}^K e^{-\nu t} \frac{(\nu t)^n}{n!} v'_n,$$

and recall that

$$H_K(t) = 1 - \sum_{n=0}^K e^{-\nu t} \frac{(\nu t)^n}{n!} \text{ and } S_K = \frac{M_K + m_K}{2}.$$

**Theorem 3.1** *For every  $t \geq 0$ , we have*

$$\left| EIAV(t) - \left[ G'_K(t) + \frac{K+1}{\nu t} (v'_K - S_K) H_{K+1}(t) + S_K H_K(t) \right] \right| \leq \varepsilon/4 \quad (9)$$

**Proof.** For every  $t \geq 0$ , we have

$$EIAV(t) = G'_K(t) + \phi(t),$$

where

$$\phi(t) = \sum_{n=K+1}^{\infty} e^{-\nu t} \frac{(\nu t)^n}{n!} v'_n.$$

For  $n \geq K+1$ , we have

$$\begin{aligned} v'_n &= \frac{1}{n+1} \sum_{k=0}^n v_k \\ &= \frac{1}{n+1} \left[ \sum_{k=0}^K v_k + \sum_{k=K+1}^n v_k \right] \\ &= \frac{K+1}{n+1} v'_K + \frac{1}{n+1} \sum_{k=K+1}^n v_k \\ &= \frac{K+1}{n+1} v'_K + \frac{1}{n+1} \sum_{k=K+1}^n (v_k - S_K) + \frac{(n-K)}{n+1} S_K \\ &= \frac{K+1}{n+1} v'_K + \frac{(n-K)}{n+1} S_K + x_n \end{aligned}$$

where

$$|x_n| = \left| \frac{1}{n+1} \sum_{k=K+1}^n (v_k - S_K) \right| \leq \frac{1}{n+1} \sum_{k=K+1}^n |v_k - S_K| \leq \frac{(n-K)}{n+1} \frac{\varepsilon}{4} \leq \varepsilon/4.$$

The inequality  $|v_k - S_K| \leq \varepsilon/4$ , for  $k \geq K$ , follows from Lemma 2.1; it has already been used to bound the error  $e_1(K)$  in Relation (4). If  $\psi(t)$  is the function defined by

$$\psi(t) = \sum_{n=K+1}^{\infty} e^{-\nu t} \frac{(\nu t)^n}{n!} x_n,$$

we obtain  $|\psi(t)| \leq \varepsilon/4$ . We then have

$$\phi(t) = \sum_{n=K+1}^{\infty} e^{-\nu t} \frac{(\nu t)^n}{n!} \left( \frac{K+1}{n+1} v'_K + \frac{(n-K)}{n+1} S_K \right) + \psi(t).$$

By writing  $(n-K) = n+1 - (K+1)$  in this last expression, we get

$$\phi(t) = \frac{K+1}{\nu t} (v'_K - S_K) H_{K+1}(t) + S_K H_K(t) + \psi(t).$$

We then obtain

$$\left| EIAV(t) - \left[ G'_K(t) + \frac{K+1}{\nu t} (v'_K - S_K) H_{K+1}(t) + S_K H_K(t) \right] \right| \leq \psi(t),$$

which completes the proof since  $\psi(t) \leq \varepsilon/4$ . ■

**Theorem 3.2** *For every  $\varepsilon > 0$ , for every  $t$  and  $t'$  such that  $t \geq t' \geq T_K$  we have*

$$\left| EIAV(t) - \left[ \frac{t'}{t} EIAV(t') + \left( 1 - \frac{t'}{t} \right) S_K \right] \right| \leq \varepsilon \quad (10)$$

**Proof.** For every  $t$  and  $t'$  such that  $t \geq t' \geq T_K$ , we have

$$\begin{aligned} EIAV(t) &= \frac{1}{t} \int_0^t PAV(s) ds \\ &= \frac{1}{t} \left[ \int_0^{t'} PAV(s) ds + \int_{t'}^t PAV(s) ds \right] \\ &= \frac{1}{t} \left[ \int_0^{t'} PAV(s) ds + (t - t') S_K + \int_{t'}^t [PAV(s) - S_K] ds \right] \\ &= \frac{t'}{t} EIAV(t') + \left( 1 - \frac{t'}{t} \right) S_K + \frac{1}{t} \int_{t'}^t [PAV(s) - S_K] ds \end{aligned}$$

Using Relation (7), we have, since  $t' \geq T_K$ ,

$$\left| \frac{1}{t} \int_{t'}^t [PAV(s) - S_K] ds \right| \leq \frac{1}{t} \int_{t'}^t |PAV(s) - S_K| ds \leq \left( 1 - \frac{t'}{t} \right) \varepsilon \leq \varepsilon,$$

which completes the proof. ■

Note that Theorem 3.2 is still valid if we replace  $T_K$  by  $\lceil T_K \rceil$ . So, as for the point availability, we can use  $\lceil T_K \rceil$  instead of  $T_K$  to make easier the computation of the expected interval availability.

Using these two theorems, we obtain a new algorithm to compute the expected interval availability which is similar to the one described in Table 2 for the point availability. It suffices to perform the following changes in the algorithm given in Table 2. The computation of  $v'_n$  given by Relation (8) must be added just after the computation of  $v_n$ , with  $v'_0 = v_0$ . The relation  $PAV(t_j) = G_K(t_j)$  must be replaced by  $EIAV(t_j) = G'_K(t_j)$  and the computations of  $PAV(t_j)$  in the case where  $K \leq N$  must be replaced by those of  $EIAV(t_j)$  given in Relation (9) for  $t_j \leq T_K$  and in Relation (10) for  $t_j > T_K$ . To use the Relation (10), we need  $EIAV(t)$  for one value of  $t$  such that  $t \geq T_K$ . Such a value can be obtained by using one more time the Relation (9) for the smallest value of  $t_j$  such that  $t_j \geq T_K$ . Note that we have the well-known stationary relation  $PAV(\infty) = EIAV(\infty)$ .

## 4 Numerical Example

We consider a fault-tolerant multiprocessor system with finite buffer stages. This system was first considered in [9] for two processors without repair and has been extended in [10] to include repair for the computation of the moments of performability. It has been also used in [11] to obtain the distribution of performability. We use here the same model for the computation of the point availability with our new method. It consists of  $n$  identical processors and  $b$  buffer stages. Processors fail independently at rate  $\lambda$  and are repaired singly with rate  $\mu$ . Buffer stages fail independently at rate  $\gamma$  and are repaired with rate  $\tau$ . Processor failures causes a graceful degradation of the system and the number of operational processors is decreased by one. The system is in a failed state when all the processors have failed or any of the buffer stages has failed. No additional processor failures are assumed to occur when the system is in a failed state. The model is represented by a Markov process with state transition diagram shown in Fig. 1. The state space of the system is  $S = \{(i, j); 0 \leq i \leq n, j = 0, 1\}$ . The component  $i$  of a state  $(i, j)$  means that there are  $i$  operational processors and the component  $j$  is zero if any of the buffer stages is failed, otherwise it is one. It follows that the set  $U$  of operational states is  $U = \{(i, 1); 1 \leq i \leq n\}$ .

We evaluate the point availability given that the system started in state  $(n, 1)$ . The number of processors is fixed to 16, each with a failure rate  $\lambda = 0.01$  per week and a repair rate  $\mu = 0.1666$  per hour. The individual buffer stage failure rate is  $\gamma = 0.22$  per week and its repair rate is  $\tau = 0.1666$  per hour. The error tolerance is  $\varepsilon = 0.00001$ .

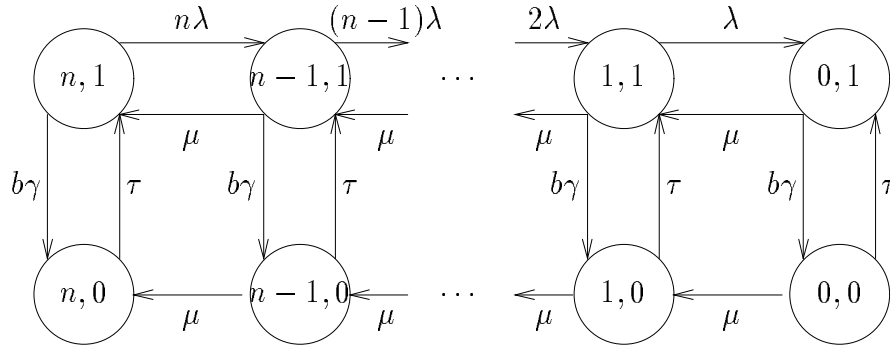


Fig. 1: State-transition diagram for a  $n$ -processor system

In Fig. 2, we plot the point availability,  $PAV(t)$ , as a function of  $t$  for different values of the number of buffer stages  $b$ . The largest value of  $t$ , that is the value of  $t_J$  in the algorithm, has been chosen equal to 10000 hours.



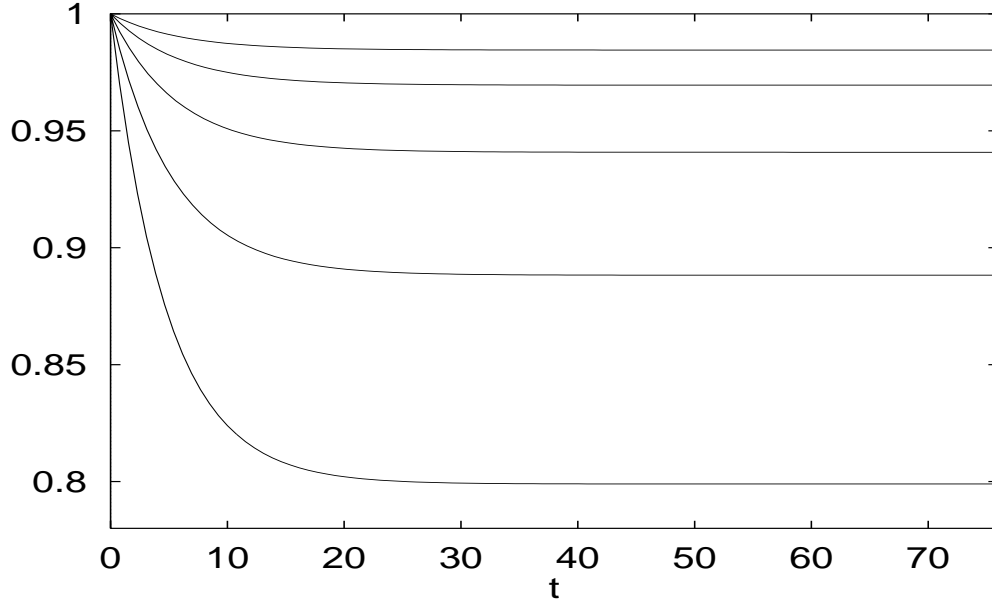


Fig. 2: From top to the bottom:  $PAV(t)$  for  $b = 2, 4, 8, 16, 32$

For that largest value of  $t$  we show in Fig. 3 the truncation step  $N = N_{10000}$ , the discrete time to stationarity  $K$  and the continuous time to stationarity  $T_K$  (in fact we give  $\lceil T_K \rceil$ ) for different values of the number of buffer stages  $b$ . This figure shows for example that when  $b = 16$  the classical algorithm needs 3581 matrix-vector products and our new algorithm needs only 18 matrix-vector products, the continuous time to stationarity being equal to 77. When  $b = 1024$  the classical algorithm needs 15616 matrix-vector products and our new algorithm needs only 86 matrix-vector products, the continuous time to stationarity being equal to 62. Moreover our algorithm also computes the steady state point availability with a precision equal to  $\varepsilon/4$ . Fig. 3 also shows that both situations,  $K < T_K$  and  $K > T_K$ , are possible.

$b$	2	4	8	16	32	64	128	256	512	1024
$N$	3581	3581	3581	3581	3581	3581	3602	5334	8776	15616
$K$	19	19	18	18	18	18	18	28	48	86
$\lceil T_K \rceil$	81	81	80	78	77	75	77	70	66	62

Fig. 3: Stationarity detection for different numbers of buffer stages

We consider in Fig. 4 smaller values of  $t_J$ . The number of buffer stages is fixed to  $b = 8$ . For  $t_J \leq 10$  we get  $N_{10} \leq 14$  and the discrete time to stationarity  $K$  is not reached. This

means that  $K > 14$ . For  $t_J \geq 20$  we get  $N_{t_J} \geq 20$  and the discrete time to stationarity is reached. Its value is  $K = 18$  and the continuous time to stationarity is  $\lceil T_K \rceil = 80$ . Fig. 4 shows that even for small values of  $t_J$  ( $t_J < T_K$ ), our algorithm can reduce the computation time with respect to the classical algorithm. For instance when  $t_J = 60$ , the classical algorithm needs 42 matrix-vector products and our new algorithm needs only 18 matrix-vector products.

$t_J$	10	20	30	40	50	60	70	80	90	100
$N_{t_J}$	14	20	26	32	37	42	47	51	56	61

Fig. 4: Stationarity detection for small values of the time.

## 5 Extension to the performability analysis

The method proposed for the computation of the point availability and the expected interval availability using the steady state availability detection can be extended to more general measures such as the point performability and the expected interval performability.

In performability modeling (see, for instance, [9, 10, 11, 12, 13, 14, 15, 16] and the references therein) reward rates are associated with states of the model to quantify the ability of the system to perform in the corresponding states. We denote by  $\rho(i)$  the reward rate associated to the state  $i \in S$ . The reward rates  $\rho(i)$  are assumed to be nonnegative real numbers. The point performability at time  $t$ , denoted by  $PP(t)$ , and the expected interval performability, denoted by  $EIP(t)$ , are defined by

$$PP(t) = \sum_{i \in S} \rho(i) \Pr\{X_t = i\} \text{ and } EIP(t) = \frac{1}{t} \int_0^t PP(s) ds.$$

We define  $\rho = \max_{i \in S} \rho(i)$  and  $r(i) = \rho(i)/\rho$  and we denote by  $r$  the column vector whose  $i$ th entry is equal to  $r(i)$ . We then have  $PP(t) = \rho f(t)$  and  $EIP(t) = \rho g(t)$ , where

$$f(t) = \alpha e^{At} r \text{ and } g(t) = \frac{1}{t} \int_0^t f(s) ds.$$

Since for every  $i \in S$ , we have  $0 \leq r(i) \leq 1$ , all the results and algorithms obtained for the computation of the availability measures can be easily extended to the computation of  $f(t)$  and  $g(t)$ . To do that it suffices to replace the column vector  $1_U$  by the column vector  $r$ . The values  $M_0$  and  $m_0$  becomes  $M_0 = \max_{i \in S} r(i)$  and  $m_0 = \min_{i \in S} r(i)$ . Moreover we have  $f(\infty) = g(\infty) = \pi r$ .

## 6 Conclusions

A new algorithm has been developed to compute the point availability and the expected interval availability of repairable computer systems modeled by Markov processes. This new algorithm is based on the uniformization technique and on the detection of the steady state availability. It compares favorably with the classical uniformization algorithm when the time horizon is large and it is shown through a numerical example that computational savings can be obtained even when the time horizon is small. Moreover our algorithm gives the steady state availability if the stationarity is reached and bounds of the steady state availability otherwise. Finally this method can be easily extended to the computation of more general measures such as the point performability and the expected interval performability.

## References

- [1] E. de Souza e Silva and H. R. Gail, "Calculating cumulative operational time distributions of repairable computer systems," *IEEE Trans. Computers*, vol. C-35, pp. 322–332, April 1986.
- [2] G. Rubino and B. Sericola, "Interval availability distribution computation," *Proceedings IEEE 23-th Fault-Tolerant Computing Symposium, Toulouse, France*, pp. 49–55, June 1993.
- [3] G. Rubino and B. Sericola, "Interval availability analysis using denumerable Markov processes: application to multiprocessor subject to breakdowns and repair," *IEEE Trans. Computers. Special issue on fault-tolerant computing*, vol. C-44, pp. 286–291, February 1995.
- [4] G. Ciardo, A. Blakemore, P. F. Chimento, J. K. Muppala, and K. S. Trivedi, "Automated generation and analysis of Markov reward model using stochastic reward nets," in *Linear Algebra, Markov Chains, and Queueing Models* (C. D. Meyer and e. R. J. Plemmons, eds.), pp. 145–191, Springer-Verlag, 1993.
- [5] M. Malhotra, J. K. Muppala, and K. S. Trivedi, "Stiffness-tolerant methods for transient analysis of stiff Markov chains," *Microelectronics and Reliability*, vol. 34, no. 11, pp. 1825–1841, 1994.
- [6] S. M. Ross, *Stochastic Processes*. John Wiley and Sons, 1983.
- [7] P. N. Bowerman, R. G. Nolty, and E. M. Scheuer, "Calculation of the Poisson cumulative distribution function," *IEEE Trans. Reliability*, vol. 39, pp. 158–161, 1990.

- 
- [8] E. Parzen, *Stochastic Processes*. San Francisco: Holden-Day, Inc, 1962.
  - [9] J. F. Meyer, "Closed-form solutions for performability," *IEEE Trans. Computers*, vol. C-31, pp. 648–657, July 1982.
  - [10] B. R. Iyer, L. Donatiello, and P. Heidelberger, "Analysis of performability for stochastic models of fault-tolerant systems," *IEEE Trans. Computers*, vol. C-35, pp. 902–907, October 1986.
  - [11] V. G. Kulkarni, V. F. Nicola, R. M. Smith, and K. S. Trivedi, "Numerical evaluation of performability and job completion time in repairable fault-tolerant systems," *Proceedings IEEE 16-th Fault-Tolerant Computing Symposium, Vienna, Austria*, pp. 252–257, July 1986.
  - [12] G. Ciardo, R. Marie, B. Sericola, and K. S. Trivedi, "Performability analysis using semi-Markov reward processes," *IEEE Trans. Computers*, vol. C-39, pp. 1251–1264, October 1990.
  - [13] R. M. Smith, K. S. Trivedi, and A. V. Ramesh, "Performability analysis: measures, an algorithm, and a case study," *IEEE Trans. Computers*, vol. C-37, pp. 406–417, April 1988.
  - [14] H. Nabli and B. Sericola, "Performability analysis of fault-tolerant computer systems," Tech. Rep. 2254, INRIA, Campus de Beaulieu, 35042 Rennes Cedex, France, May 1994. To appear in *IEEE Trans. Computers*.
  - [15] E. de Souza e Silva and H. R. Gail, "Calculating availability and performability measures of repairable computer systems using randomization," *J. ACM*, vol. 36, pp. 171–193, January 1989.
  - [16] J. F. Meyer, "On evaluating the performability of degradable computing systems," *IEEE Trans. Computers*, vol. C-29, pp. 720–731, August 1980.



---

Unité de recherche INRIA Lorraine, Technopôle de Nancy-Brabois, Campus scientifique,  
615 rue du Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY  
Unité de recherche INRIA Rennes, Irisa, Campus universitaire de Beaulieu, 35042 RENNES Cedex  
Unité de recherche INRIA Rhône-Alpes, 655 avenue de l'Europe, 38330 MONTBONNOT ST MARTIN  
Unité de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex  
Unité de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

---

Éditeur  
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)  
ISSN 0249-6399